AU 332 Artificial Intelligence: Principles and Techniques
Homework 3 Due NOV 2nd 11:59pm

Adhere to the Code of Academic Integrity. You may discuss background issues and general strategies with others and seek help from course staff, but the implementations that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is never OK for you to see or hear another student's code and it is never OK to copy code from published/Internet sources. If you feel that you cannot complete the assignment on you own, seek help from the course staff.

1 Reinforcement Learning

This exercise will familiarize you with Reinforcement Learning and its use to search for treasure. Please finish the homework individually.

1.1 Required Problem

In this assignment you need to write a program that will learn to find the optimal policy of a map given below.
Game description: Let's look at a simple scenario, a mouse is trying to get to a piece of cheese marked in dark red circle. Additionally, there is a cliff in the map that must be avoided, or the mouse falls, gets a negative reward, and has to start back at the beginning. The cliff is marked in dark blue circles. The simulation looks something exactly like the image shown in Figure 1. The black squares are the walls of the map.
We model the game as a non-deterministic MDP. The action has 80% of the time moving to the direction it wants to and 10% of the time moving to the left or the right of the desired direction.

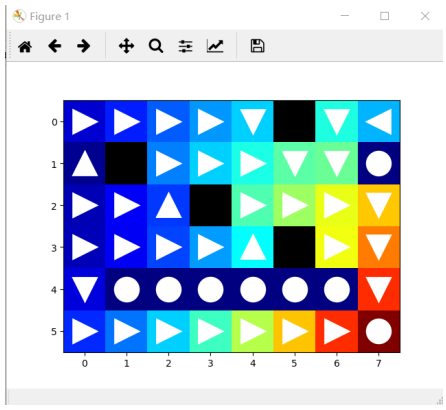

Figure 1: A mouse in the maze

(i) Task 1: Implement **run_policy_iterations** in the **gridworld.py** using policy iteration.

(ii) Task 2: Implement Q-Learning and SARSA, and compute an optimal policy against the given map shown in Figure 1. For both algorithms, use ϵ-greedy for exploration. In your report, submit an output of the optimal action for each state. You need to change *qlearn.py* and *sarsa.py* for this task.

(iii) Task 3: implement exploration function for Q-learning and show us how this methods can reduce regret comparing to your earlier implementation of ϵ-greedy.

(iv) Task 4: Try to improve the results by modifying the parameters including γ, α, ϵ for the Q learning algorithm. In your report, show us how you tuned those hyper-parameters and what are the improved results from your choices of the parameters.

(v) You need to submit *qlearn.py*, *sarsa.py*, and *gridworld.py*.

1.2 Bonus problem

Gym is a toolkit for developing and comparing reinforcement learning algorithms.(http://gym.openai.com/)
It supports teaching agents everything from walking to playing games like Pong or Pinball. Now you are required to design an AI to reach high scores in one of your favorite Atari games.(http://gym.openai.com/envs/#atari)

```
import gym
env = gym.make('Pong-v0')

for i_episode in range(100):
    env.reset()
    for t in range(100):
        env.render()
        action = env.action_space.sample()
        observation, reward, done, info = env.step(action)
        if done:
            print("Episode finished after {} timesteps".format(t+1))
            break
```

(i) Pick a gym environment and understand its build in methods. Including the state space, the action space and the reward space.

(ii) Improve your Q-learning algorithm and adjust it to play the atari game for the gym environment you selected.

(iii) You are required to submit a python file named *gym.py* and *Q_learning_gym.py*. Write a short summary of how you improved Q learning to achieve better results.

2 Submission instructions

1. Zip all your python files and `HW3.pdf` to a folder called $homework3_name.zip$
2. Send the zip file to TA 121103451@*qq.com*